In [108…
```
import numpy as np
import pandas as pd
from sklearn.svm import SVR
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt
plt.style.use('classic')
```

In [109…
```
filepath=r"C:\Users\AKASH DEEP\Downloads\BTC-USD (1).csv"
df=pd.read_csv(filepath)
df
```

Out[109…

	Date	Open	High	Low	Close	Adj Close	Volume
0	44197	28994.00977	29600.62695	28803.58594	29374.15234	29374.15234	4.073030e+10
1	44198	29376.45508	33155.11719	29091.18164	32127.26758	32127.26758	6.786542e+10
2	44199	32129.40820	34608.55859	32052.31641	32782.02344	32782.02344	7.866524e+10
3	44200	32810.94922	33440.21875	28722.75586	31971.91406	31971.91406	8.116348e+10
4	44201	31977.04102	34437.58984	30221.18750	33992.42969	33992.42969	6.754732e+10
…	…	…	…	…	…	…	…
73	44270	59267.42969	60540.99219	55393.16406	55907.19922	55907.19922	6.641937e+10
74	44271	55840.78516	56833.17969	53555.02734	56804.90234	56804.90234	5.974980e+10
75	44272	56825.82813	58969.81641	54528.62891	58870.89453	58870.89453	6.025831e+10
76	44273	58893.07813	60116.25000	54253.57813	57858.92188	57858.92188	5.574604e+10
77	44274	57850.44141	59498.37500	56643.70313	58346.65234	58346.65234	4.906387e+10

78 rows × 7 columns

In [110…
```
x=df.loc[:,'Date']

y=df.loc[:,'Adj Close']
```

In [111…
```
x_train, x_test, y_train, y_test = train_test_split(x, y,test_size=0.33, random_state=1
```

In [112…
```
days=list()
adj_close=list()
```

In [113…
```
for day in x_train:
    days.append([int(day)])
```

In [114…
```
for adj_close_price in y_train:
    adj_close.append(float(adj_close_price))
```

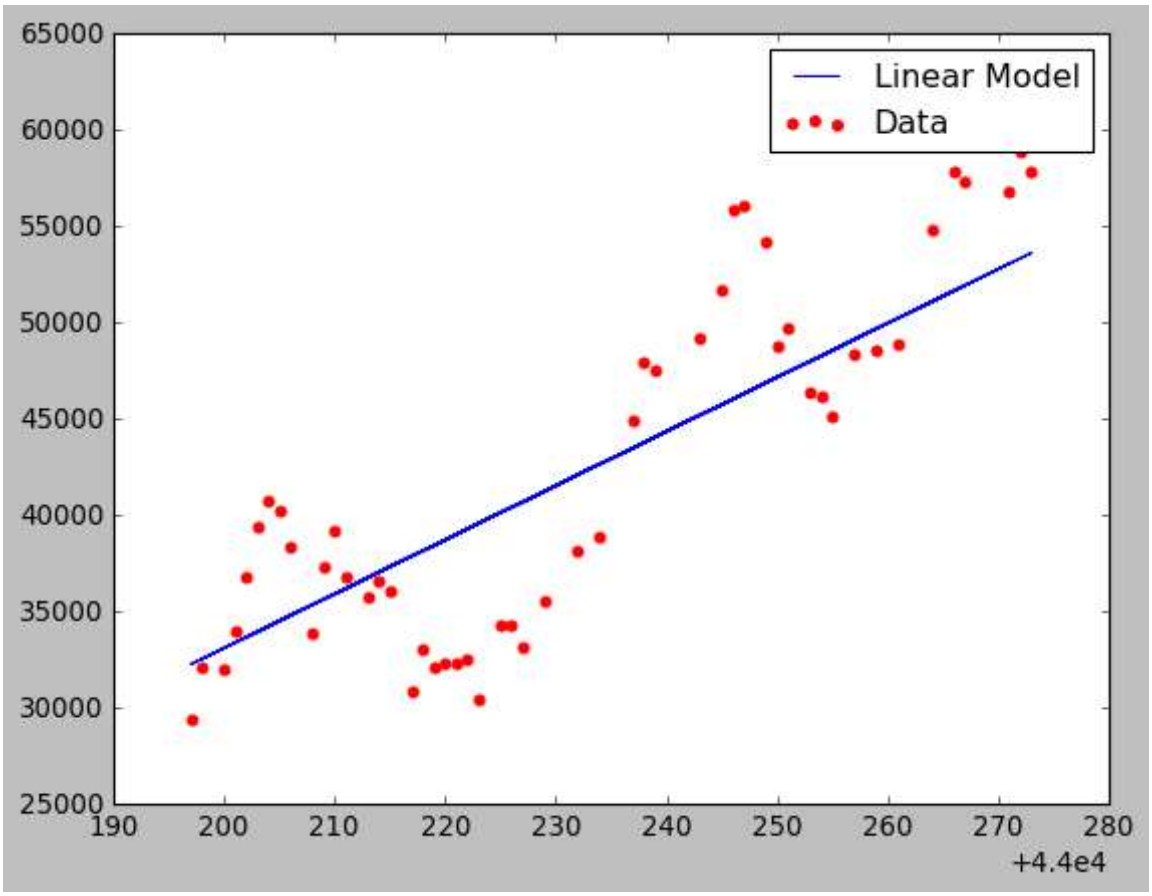
In [115…
```
print(days)
print(adj_close)
```

```
[[44223], [44273], [44272], [44259], [44237], [44254], [44232], [44218], [44266], [4420
0], [44243], [44249], [44245], [44250], [44221], [44205], [44214], [44197], [44267], [44
255], [44204], [44271], [44251], [44210], [44219], [44229], [44227], [44246], [44238],
[44253], [44220], [44201], [44264], [44211], [44226], [44225], [44239], [44257], [4420
8], [44215], [44217], [44247], [44222], [44203], [44198], [44213], [44261], [44202], [44
206], [44269], [44209], [44234]]
[30432.54688, 57858.92188, 58870.89453, 48561.16797, 44918.18359, 46188.45313, 38144.308
59, 33005.76172, 57805.12109, 31971.91406, 49199.87109, 54207.32031, 51679.79688, 48824.
42578, 32366.39258, 40254.54688, 36630.07422, 29374.15234, 57332.08984, 45137.76953, 407
97.60938, 56804.90234, 49705.33203, 39187.32813, 32067.64258, 35510.28906, 33114.35938,
55888.13281, 47909.33203, 46339.76172, 32289.37891, 33992.42969, 54824.11719, 36825.3671
9, 34269.52344, 34316.38672, 47504.85156, 48378.98828, 33922.96094, 36069.80469, 30825.6
9922, 56099.51953, 32569.84961, 39371.04297, 32127.26758, 35791.27734, 48912.38281, 3682
4.36328, 38356.44141, 59302.31641, 37316.35938, 38903.44141]
```

In [116…
```
lin_svr=SVR(kernel='linear',C=1.0)
lin_svr.fit(days,adj_close)
```

Out[116… SVR(kernel='linear')

In [117…
```
plt.scatter(days,adj_close,color='red',label='Data')
plt.plot(days,lin_svr.predict(days),color='blue',label='Linear Model')
plt.legend()
plt.show()
```



In [118…
```
day_test=list()
adj_close_test=list()
```

In [119…
```
for day in x_test:
    day_test.append([int(day)])
```

In [120…
```
for adj_close_price in y_test:
    adj_close_test.append(float(adj_close_price))
```

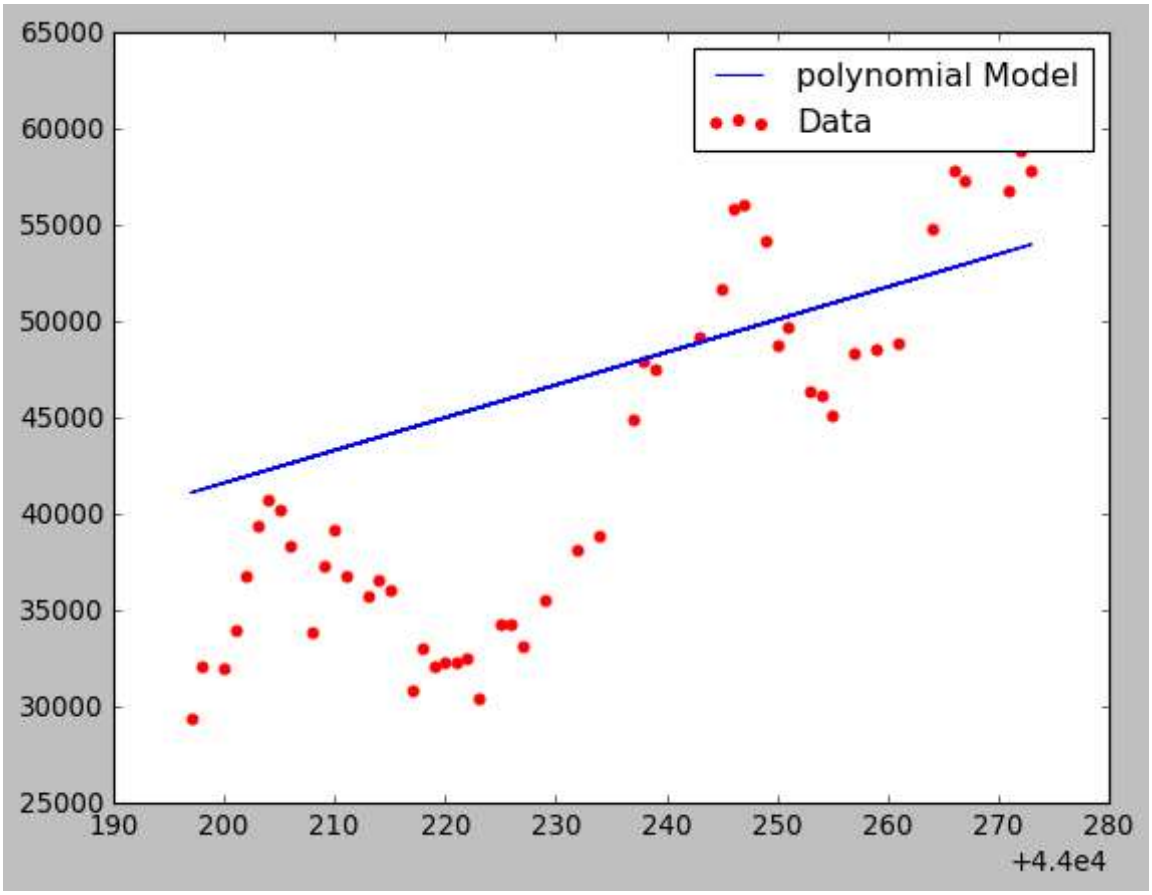
In [121…
```
accuracy= lin_svr.score(day_test,adj_close_test)
print('Accuracy with linear kernel: %.3f'%(accuracy*100))
```

```
Accuracy with linear kernel: 72.226
```

In [122…
```
#svr polynomial kernel
poly_svr=SVR(kernel='poly',C=0.01,degree=2)
poly_svr.fit(days,adj_close)
```

Out[122…
```
SVR(C=0.01, degree=2, kernel='poly')
```

In [123…
```
plt.scatter(days,adj_close,color='red',label='Data')
plt.plot(days,poly_svr.predict(days),color='blue',label='polynomial Model')
plt.legend()
plt.show()
```



In [124…
```
accuracy= poly_svr.score(day_test,adj_close_test)
print('Accuracy with polynomial kernel: %.3f'%(accuracy*100))
```

```
Accuracy with polynomial kernel: 44.547
```

In [125…

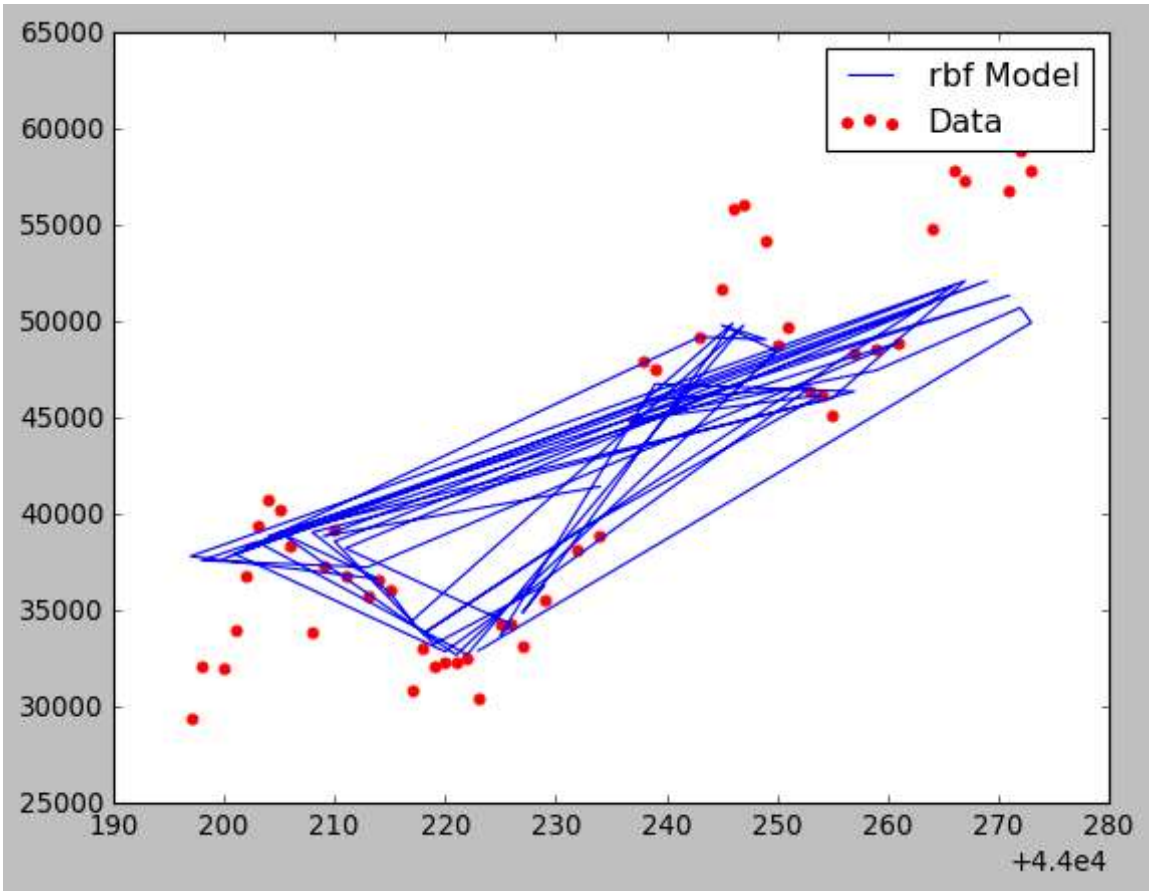
```
#svr rbf kernel
rbf_svr=SVR(kernel='rbf',C=1500.0,gamma=0.02)
rbf_svr.fit(days,adj_close)
```

Out[125…

```
SVR(C=1500.0, gamma=0.02)
```

In [126…

```
plt.scatter(days,adj_close,color='red',label='Data')
plt.plot(days,rbf_svr.predict(days),color='blue',label='rbf Model')
plt.legend()
plt.show()
```



In [127…

```
accuracy= rbf_svr.score(day_test,adj_close_test)
print('Accuracy with rbf kernel: %.3f'%(accuracy*100))
```

```
Accuracy with rbf kernel: 80.413
```